# Manual Paso a Paso: Mejora de Imagen con Python y Google Colab

# 1. Introducción

Este manual describe paso a paso cómo aplicar técnicas de mejora de imagen en imágenes en escala de grises, especialmente útiles para condiciones de baja iluminación como las del conjunto de datos The Dark Face. Utilizaremos Python en Google Colab, lo que permite realizar todo el procesamiento sin necesidad de instalaciones locales.

# 2. Requisitos

• Cuenta de Google para usar Google Colab  
• Imagen en formato JPG o PNG, preferiblemente en escala de grises  
• Conexión a internet

# 3. Crear y Abrir un Notebook en Google Colab

1. Ve a https://colab.research.google.com  
2. Crea un nuevo cuaderno ('Archivo > Nuevo cuaderno')  
3. Cambia el nombre a 'Mejora\_Imagen\_Colab' o similar

# 4. Subir Imagen e Instalar Librerías

Ejecuta la siguiente celda en Colab:

!pip install opencv-python-headless  
  
from google.colab import files  
import cv2  
import numpy as np  
from matplotlib import pyplot as plt  
from google.colab.patches import cv2\_imshow  
  
uploaded = files.upload()  
filename = list(uploaded.keys())[0]  
img = cv2.imread(filename, cv2.IMREAD\_GRAYSCALE)  
cv2\_imshow(img)

# 5. Corrección Gamma

Esta técnica aclara sombras sin saturar zonas brillantes.

def gamma\_correction(image, gamma):  
 inv\_gamma = 1.0 / gamma  
 table = np.array([(i / 255.0) \*\* inv\_gamma \* 255 for i in range(256)]).astype('uint8')  
 return cv2.LUT(image, table)  
  
gamma\_img = gamma\_correction(img, 0.4)  
cv2\_imshow(gamma\_img)

# 6. Ecualización del Histograma

Mejora el contraste general redistribuyendo los niveles de gris.

equalized\_img = cv2.equalizeHist(img)  
cv2\_imshow(equalized\_img)

# 7. Ecualización Adaptativa (CLAHE)

Mejora el contraste en zonas específicas sin sobreexponer otras.

clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8, 8))  
clahe\_img = clahe.apply(img)  
cv2\_imshow(clahe\_img)

# 8. Estiramiento de Contraste

Expande el rango dinámico de los niveles de gris.

def contrast\_stretching(image):  
 min\_val = np.min(image)  
 max\_val = np.max(image)  
 return ((image - min\_val) / (max\_val - min\_val) \* 255).astype('uint8')  
  
stretched\_img = contrast\_stretching(img)  
cv2\_imshow(stretched\_img)

# 9. Multiplicación por Escalar

Aclara toda la imagen multiplicando los valores de los píxeles.

bright\_img = cv2.convertScaleAbs(img, alpha=1.3, beta=0)  
cv2\_imshow(bright\_img)

# 10. Promedio con Suavizado Gaussiano

Mezcla la imagen original con una suavizada para reducir ruido.

blurred = cv2.GaussianBlur(img, (5,5), 0)  
average\_img = cv2.addWeighted(img, 0.5, blurred, 0.5, 0)  
cv2\_imshow(average\_img)

# 11. Guardar y Descargar Imágenes

Puedes guardar y descargar las imágenes procesadas con:

cv2.imwrite('img\_gamma.jpg', gamma\_img)  
from google.colab import files  
files.download('img\_gamma.jpg')

# 12. Recomendaciones Finales

• Compara visualmente los resultados de cada técnica.  
• Inserta las mejores versiones en tu informe.  
• Ajusta parámetros como gamma o clipLimit según la imagen.